Association for Information Systems
AIS Electronic Library (AISeL)

MCIS 2015 Proceedings

Mediterranean Conference on Information Systems (MCIS)

2015

Designing of Adaptive Behaviour

Dov Te'eni
Tel Aviv University, Israel, teeni@post.tau.ac.il

DESIGNING FOR ADAPTIVE BEHAVIOUR

Complete Research

Te'eni, Dov, Tel Aviv University, Israel, teeni@tau.ac.il

Abstract

We need, increasingly, to design systems that can support complex behaviours in which users *adapt* dynamically to the progression of the task or to changing conditions during the interaction. Unfortunately, we lack procedures and methods to build systems for adaptive behaviour. This article proposes an approach that identifies dimensions on which users adapt their behaviour and then determines the corresponding design implications on how the system should adjust to fit the user's adaptive behaviour.

Keywords: Adaptive behaviour, Levels of abstractions, Human-computer interaction, Design.

1 Introduction

Increasingly, we need to design the human-computer interface of systems that support complex tasks in which users *adapt* dynamically to the progression of the task or to changing conditions during the interaction. There is abundant research on adaptive systems that adapt to the user's characteristics and preferences (Billsus et al., 2002) or to the conditions at the time of use. There is also research on how users adapt their use of systems, e.g., using more features of the system (Sun, 2012). Unfortunately, we lack procedures and methods to build systems that support the user's adaptive behaviour whilst performing a task. This article proposes an approach that identifies dimensions on which users adapt their behaviour and then determines the corresponding design implications on how the system should adjust to fit the user's adaptive behaviour. Future work will examine technical solutions based on advanced technologies for data acquisition and analytics but for now, in this first step, we demonstrate our approach with extant technologies.

A reader of a professional article, such as this article, may read it in at least two different modes. The reader can scan the article by looking at prominent pieces of information such as the titles, the first sentence of a paragraph, anything in bold or italics, tables and graphs. A second mode of reading would be to read serially consecutives pages, attempting to read sequentially large parts of the text. Most readers, start, out of habit, reading an online article (or an information webpage) in the first mode. An adaptive reader seeking, say, a practical question about designing an interface may toggle between the modes, depending on the relevance of the content to the question in mind. Moving from one mode to another is not always obvious or simple and is often triggered by external situations, but once in a certain mode, the reader reads, out of habit, in a routine that is quite automatic. This simple and common example demonstrates a dimension of reading specificity (high-level reading versus detailed-level reading) on which the reader moves up and down the levels according to considerations of speed and convenience and the likelihood of success in finding an answer.

Assuming that such moves between reading modes are advantageous, there are at least two broad design implications. One implication is the need to design a system that supports an easy transition between reading modes. A second implication is to fit the system to best serve each reading mode in its turn. For instance, when scanning the article in the high-level reading mode, the reader essentially ignores non-prominent pieces of information according to a principle called visual hierarchy (Djamasbi et al., 2011), letting them almost disappear into the background. A commonly used technique to emphasize parts of the text and downplay surrounding texts is to present the page in a fisheye view.

A third, more complex design implication of the two reading modes is to prompt the user to move effectively between modes. To do so, the system would have to rely on knowledge of the relative effectiveness of one mode versus the other, given a particular task such as finding in the article a practical design solution. For instance, knowing the reader is looking for a practical solution, the system could suggest a move to detailed-level reading when the reader gets closer to the appropriate section in the article (like a car dashboard suggesting a change of gear).

In practice, not enough is being done to design for adaptive behaviour, in part perhaps, because we need to know why and how people adapt when performing specific tasks. Furthermore, adaptive behaviour relies on complex feedback. Most interactive systems provide simple feedback to users immediately upon the user's action such as 'Input received'. In contrast, this paper talks about longer cycles of feedback such as feedback throughout a series of steps needed to complete a flight reservation. This requires looking at cognitive, affective and social aspects of thinking and communicating. We refer here to adaptive behaviour that may include, for instance, moving up and down levels of abstraction, changing moods in emotional activities, and oscillating between levels of networking. How do you support effectively such adaptive behaviour?

We propose a procedure of five steps to design systems that support adaptive behaviour:

1. Identify dimensions for adaptive behaviour in a given activity
2. Determine the consequences of transitions between states on a dimension
3. Design systems to support transitions between states
4. Fit the system to the state to which the user moved
5. Design systems that guide advantageous behaviour

The remainder of the paper develops these ideas systematically. The next section reviews the idea of fitting, statically, the human-computer interface to characteristics of the user and the task. A subsequent section, extending the static view of fit to a dynamic view, develops the above five-step procedure. It demonstrates the approach by examining two distinct cases of dimensions of change. One has to do with individuals adapting throughout a modelling session and the other colleagues adapting their patters of interaction throughout a project. The final sections talk about extensions, implications and current developments.

2 Fit in human-computer interaction

There is substantial and expanding research on fitting the interface according to the user's characteristics or preferences, according to the purpose of using the system and according to the context in which it is used. Figure 1 shows a framework for studying 'fit' in goal-oriented human-computer interaction in which the user's goal is to accomplish a task (Te'eni, 2006). The framework takes the perspective of an individual user. In today's world of collaboration and social computing, this approach should be expanded to collective human-computer interaction too. The three main components of the framework are user, task and computer, and the fit among them can be operationalized as three human aspects, namely physical, cognitive and affective. In a collaborative context, social fit should also be considered. Extant research has shown, in particular, that the interaction of cognition with task and computer depends on whether behaviour is habitual or mindful. This section looks primarily at habitual tasks, such as reading or following some pre-defined rule such as normalizing a data table, but not at creative tasks, such as generating a new product. A good fit among the three components is expected to lead to better performance and well-being for the user. Intuitively, a good fit is one that minimizes the effort required to perform the task, be it physical effort, such as hand movement, or cognitive effort, such as the need to translate a graph to numbers or vice versa (see for example the work on cognitive fit by Vessey and Galletta, 1991).

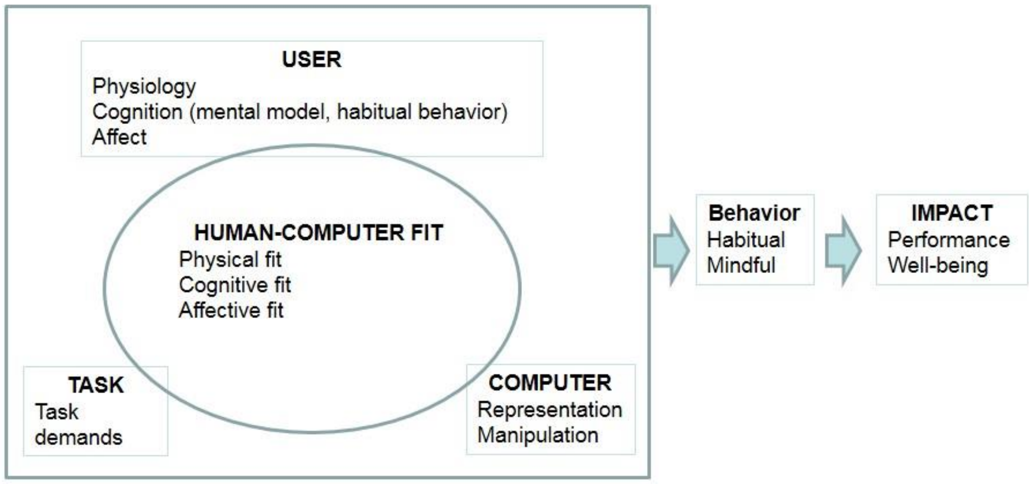


Figure 1. *A framework for studying fit for an individual user (adapted from Te'eni, 2006).*

Figure 1 suggests several types of human-computer fit according to the combinations of factors in a given context. For example, designers attempt to fit the screen brightness to the current conditions of light intensity (physical environment) and fit the fonts to the user's age and taste (the user's sensory capacity but also the user's preferences). These examples of fit are straightforward and rely on common knowledge and experience. However, less obvious instances of fit will have to rely on knowledge about what constitutes a good fit under particular circumstances. For instance, fitting the right format of a quantitative graphic will depend on knowledge of the task, of the graphics and of the user's cognitive capacity (Kennedy et al., 1998). Other more complex examples include 1) fitting the level of detail in a message, which would depend on the communication complexity experienced by the user, and 2) fitting the tone and facial expressions of avatars to the user's mood, which would depend on psychological knowledge of emotions. Clearly, more research is needed to determine how to adapt the interface for almost unlimited combinations of users, tasks and contexts.

The next section concentrates on one aspect of adaptation, namely adaptive behaviour during a session of human-computer interaction that involves moving between different modes of behaviour. The move between modes calls for dynamic fit rather than the static fit described above, i.e., at any point of time, it is necessary to fit the interface to the current mode of behaviour. As behaviour in each mode is assumed to be routine or well defined according to known rules, good fit will usually imply minimal interference with the expected behaviour. In contrast, the moves between modes require more complex support as shown below.

3 Dimensions of adaptive behaviour for dynamic fit

Each of the fit examples above may be regarded as a one-time adaptation, say, at the beginning of a work session with the computer. Alternatively, the system could be readapted, once the conditions change, even amidst a session. A simple example is fitting the screen brightness to the prevailing light intensity; the screen brightness can readjust automatically when the light intensity changes. Adaptation during a session is more complicated when it involves changes in the user's behaviour. For instance, when the user communicating online with a friend needs to adapt the communication during a session in order to avoid misunderstandings, it is not clear how to adapt. The user, for instance, could increase the level of message detail and provide fuller explanations, which would likely reduce errors and misunderstanding but not if the problem is the language. To understand when a user should increase message details and how the online system should be designed accordingly, we need a systematic procedure for supporting adaptive behaviour that builds on knowledge of communication.

The five step procedure mentioned in the introduction is now developed using two examples. The first example of adaptive behaviour is taken from data modelling.

1. Identify dimensions for adaptive behaviour

The first step is to determine the user's adaptive behaviour on one or more dimensions. Behaviour adapts during the activity of performing the task and therefore the activity delineates the time-related scope of adaptation. In the case of data modelling, we assume, for simplicity, that the activity can be completed in one human-computer session (stretching the activity over several sessions complicates somewhat the design).

When building, say, a class diagram of a system, data modellers adapt their modelling behaviour on at least two dimensions. One dimension is the levels of abstraction at which the data modellers think, and the second dimension is the type of activity in which they engage, e.g., planning, scoping and testing (Srinivasan and Te'eni, 1995). Figure 2 shows the transitions of a subject on both dimensions (between levels of abstraction 0-4 and between activities A-D). The figure was established based on a protocol of a think-aloud session to solve a data modelling exercise. In combination with a measure of performance in the data modelling exercise, the pattern of moves on the two dimensions represents the knowledge base for designing for adaptive behaviour in modelling. We will concentrate on levels of abstraction only, which is highly applicable to most forms of human-computer interaction. Concretely, in data modelling, one could think of entities as one level of abstraction, properties of the entity as a lower level and clusters of entities (e.g., a vehicle is a cluster of car and truck) as a higher level. Adaptive behaviour is working on one level, say entities, and climbing to a higher level of clusters when progress loses direction or dropping to a lower level of properties when modelling or testing fails.

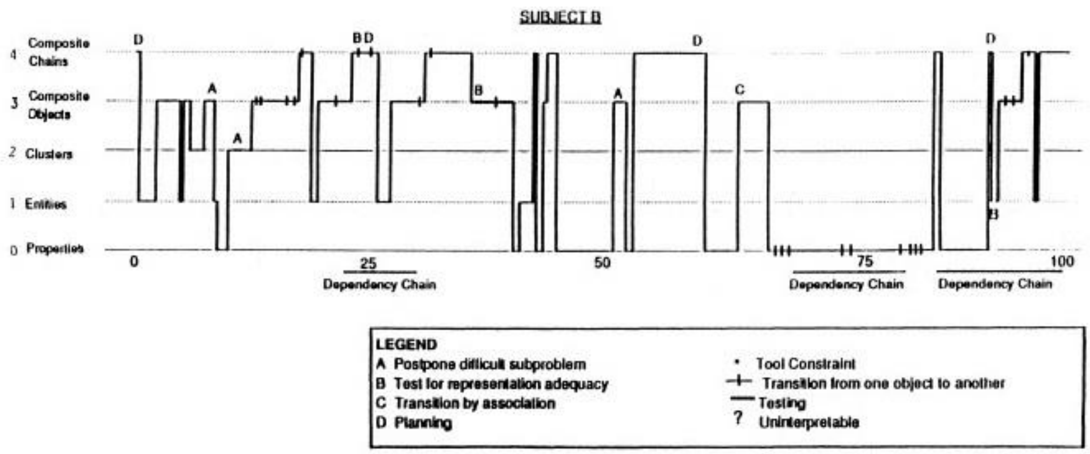


Figure 2: Adaptive behaviour when data modeling (from Srinivasan and Te'eni, 1995).

2. Determine the consequences of transitions between states on a dimension

Having identified the dimension of transitions that characterizes user behaviour, we ask what, if at all, are the consequences of certain patters of transitions. Knowledge of data modelling suggests that certain patterns are more effective than others in problem solving. For instance, working mostly at lower levels of abstraction was less effective than working at higher levels, although working only at higher levels of abstraction was ineffective. An effective pattern was to work at the level of entities but from time to time climb to the higher levels for short periods. There are other effective patterns, which may depend on the user's level of expertise (Srinivasan and Te'eni, 1995). In any event, in this example and on the basis of knowledge about the specific phenomenon of data modelling, it is possible to determine the relationship between patterns of transitions and consequences. In other cases, we may be able to define a dimension of adaptation but we may not be able to determine the consequences of alternative patterns.

3. Design systems to support transitions between states

The most obvious design implication, and yet one that more often than not is ignored in practice, is to enable easy transition between levels of the dimension identified. In the case of levels of abstraction, this would mean, for example, an easy transition between entities and clusters of entities. If we know that users tend to move from one level to another on a dimension, the system should be design to support the transitions even if the consequences of transitions is not known (at least if it is not known to be detrimental). Easy transitions between levels of abstraction means both easily operating the system to reveal and move the focus of attention to another level and supporting cognitively the transition (Sun, 2012). The move from one level to another usually means breaking away from the habitual behaviour into which the user has settled in and requiring the user actively think how to proceed, and this requires some new and forceful condition or some external trigger (Louis and Sutton, 1991).

Technically effecting the transition to another level would usually involve some form of direct manipulation such as a single click to zoom in or an option to hover over an entity, reveal its properties and move to one of them. Cognitively supporting the transition between levels includes at least two types of support: underscoring the new state as the current focus of attention and maintaining the source as context when arriving at the target. In other words, the user often needs first to realize that the focus has shifted to the new level of properties and at the same time to see the level of entity as the context for working on the properties. For example, in Figure 3, the user has moved from the working on entities to focusing on the properties of a particular entity labeled 'Person.ContactType'. The small map of entities (the higher level of abstraction from which the user moved to the current focus) is left on the screen to present the higher level as the context for the current focus. At any time, the user can go back to the higher level by clicking on the small map.

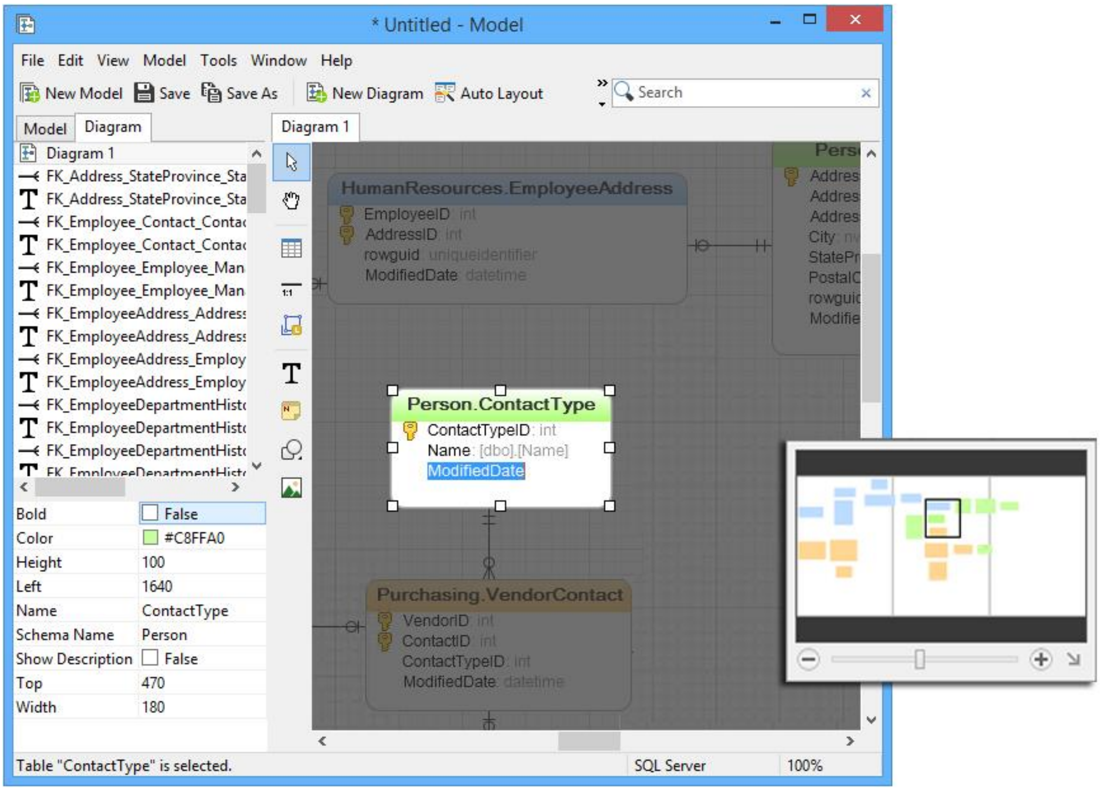


Figure 3: A screen for working at the properties level of abstraction.

4. Fit the system to the state to which the user moved

The rectangle describing the entity Person.ContactType fits the task the user is expected to perform, namely determine and check the properties of the entity, part of which are detailed in the lower left corner (e.g. Bold is set to False). A better fit would be to move the table of properties from the lower left corner closer to the central rectangle so that any moves between two require minimal effort. In fact, a clever design might be able to integrate the two in order to bring the manipulations on the entity's properties are direct as possible.

Designing for habitual behaviour differs from designing for novel behaviour that requires mindful planning and controlling. When fitting the human-computer interface to the user's expected behaviour, the designer's goal is to minimize the user's effort in performing the task correctly. When fitting the human-computer interface in novel situations, the designer's goal, in addition to minimal effort and accuracy, is to support generating new behaviours and controlling unpracticed behaviours. In the case of setting or manipulating an object's properties, behaviour is assumed to be habitual (unless the user is in training). The screen is designed to allow easy manipulations of data, immediate feedback (e.g., setting the color immediately shows the color to be displayed during use under the anticipated conditions of use), and easy and therefore more accurate detection of properties and their values.

When working at the higher level of abstraction, namely at the level of entities, the user is expected to plan the relationships between entities. The small map of entities, in an expanded form, would become the centrepiece of the screen. Its graphic presentation is a better (cognitive) fit than say a table of relationships because it lends itself most immediately to the way people represent relationships in their mind (Vessey and Galletta, 1991).

5. Design systems that guide advantageous behaviour

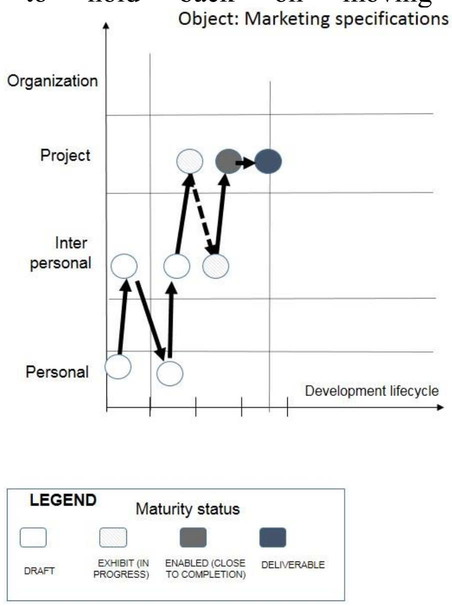
The last of the five steps in the procedure for designing adaptive behaviour systems requires knowledge of performance measures in the particular domain of the user's work. Furthermore, this step is feasible only if there is a convincing argument for affecting consequences advantageously by manipulating the human-computer interaction (step #2 above). The research quoted above argues that certain patterns of transitions are more effective than others are. For instance, remaining too long at the lowest levels of abstraction without occasionally taking a more comprehensive view by climbing to a higher level of abstraction will lead to errors. The system can detect relatively long periods of working on properties by monitoring the user's manual inputs to the system (in other cases, the system could monitor the user's gaze). The system can then alert, suggest, or force corrective action such as moving to a higher level. In other cases, it may be important to consider not only the proportionate time spent at different levels but also the sequence of activities.

Technologies that monitor users' behaviour and conditions are becoming ubiquitous in certain domains such as health and work safety but will most likely spread to many other domains of life. The nearly constant accessibility to devices such as the cell phone, online watch and wearable devices in general coupled with knowledge on what is effective in which conditions makes it feasible to guide advantageous behaviour. A simple example is RunKeeper's (an App to plan and monitor jogging activities) real time health graph, which could easily be supplemented with alerts on when you should accelerate to meet your running goal or slow down to maintain your health. Similarly, a modelling system could, in addition to signalling when the user should move to another level of abstraction, motivate the user with graphical depictions of successful patterns vis. a vis. her own actual pattern.

4 An organizational dimension of adaptive behaviour

The second example of using the procedure demonstrates a very different situation in which the dimension of adaptive behaviour is in the pattern of collaboration. The context is innovation where project members collaborate in order to develop a new product. The time dimension is not during a session but rather during stages of a project. Nevertheless, the same principles of designing for adaptive behaviour apply as we can see in the five steps of the procedure for designing adaptive behaviour systems.

Dimension of transitions. In studying how team members work on developing new products, research has shown that people work individually, in small and trusted inter-personal groups, in formal project teams and in the set of all organizational actors that are relevant to the project. Different actors at different stages of product development in different projects will show different patterns of moving from one mode to another. Figure 4 shows a particular pattern of transitions between modes when developing the marketing specifications of a new product (Merimond et al., 2012). The reasons for moving from one mode to another vary. For instance, when innovators feel they need feedback, they move from personal to inter-personal thinking but they may be reluctant to share with the entire project team for lack of trust. Innovators seem to change their patterns of behaviour depending on the organizational climate of safety. When people feel safe, they tend to move faster to wider circles of thinking but tend to hold back on moving to the project level when they feel un-



safe.

Figure 4: Innovating at differnt thinking and communicating (adapted from Merimond et al., 2012).

Consequences of patterns of transitions. Certain patterns of transitions between modes result in poor innovation or in delayed innovation. For example, when development continues at lower levels without accessing the project level early in the process, some perspectives may be ignored resulting in incomplete designs that later need to be corrected. In contrast in situations of high technological novelty, insufficient oscillations between personal and interpersonal modes results in premature formal discussions. In Figure 4, the maturity status of an object at any moment show how complete and accurate is the marketing specification. A relatively small number of oscillations between personal and interpersonal modes before the maturity status turns to the mature status ('enabled' or 'deliverable') is likely to result in poorer quality plans or products.

Design systems to support transitions. Google+ used circles to allow easy sharing of information between different groups of people regardless of what media was used. For such a system to work it, the transitions between working alone to working with a small group and between working with a small group to working with a bigger group must be easy for the user. A user who works on some object, like a marketing specification, should not be made to copy and edit it before sharing it. The system should enable a simple designation of what and with whom to share any object at any stage as a basis for communication and collaboration. Nevertheless, in some cases, the user shares information, tailoring the message to the receiving party and annotating the document with specific requests for feedback. At the same time, the system should let the user bring the discussion back to smaller circles for further 'closed' deliberations using new versions of the marketing specification.

Fit the system to the state to which the user moved. The human-computer interface for the personal mode of thinking would differ that of the interpersonal mode. In the former, the view of the marketing specification document is meant to encourage creative and critical thinking of a single user. In the interpersonal view, while the marketing specification must remain the focal object on the screen, it is necessary to add the perspectives of the interpersonal circle. For instance, it will first be important to specify who will receive the information. An example of good fit would be to show the members of the interpersonal group with pictures and names. This is the format users usually represent their close colleagues. More sophisticated interfaces help the user tailor the message to recipients, such as highlighting certain aspects that are relevant only to the particular recipient.

Design systems that guide advantageous behaviour. In this case it was difficult to propose interface designs that prompt effective patterns or discourage ineffective patterns. While it was straightforward to design easy transitions governed by the user, triggering transitions seemed risky. Our conclusion was to alert the user in two situations: when there is only one conversation at the interpersonal mode when the maturity level increase to 'close to completion' and when there are no oscillations back to individual and interpersonal modes from project level before going to the final level of maturity.

5 Beyond cognitive support

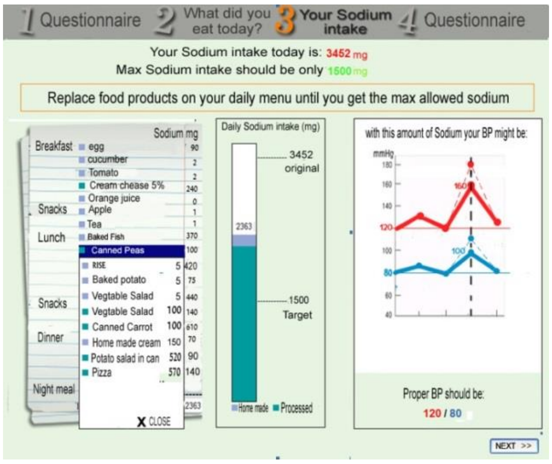


Figure 5: Personalized screen on the basis of user information (from Ronen and Te'eni, 2013).

A revealing example of the impact of personalization (adapting to user information) is in a recent study of health systems. The system supports self-management by providing ongoing feedback showing the effect of food consumption on health. The study compared personalized versus generalized information with respect to the users' attitudes and behaviour. Subjects receiving the personalized feedback (Figure 5) reported a more positive attitude and a greater propensity to follow the system's recommendations in order to improve their well-being. These findings suggest that fit is not only a means for improving the human-computer interaction by minimizing the effort to act out intended behaviour but also serves to change attitudes and behaviour.

In future, adaptive behaviour may be supported so as to achieve affective fit as well as cognitive and physiological fit. However, this would require monitoring emotions, which is not an easy task.

6 Conclusion and technological feasibility

The technological feasibility of supporting adaptive behaviour in human-computer interaction is increasing dramatically, and so will the expectations. As demonstrated above, in order to proactively recommend or even trigger adaptive behaviour, the system must recognize the user's current position on dimensions of change and the current conditions. More and more sophisticated technologies (e.g., the Internet of Things) will be available to detect information about the user, the task and the setting. For instance, a user engaged in collaborating with remote others could be notified on the susceptibility of miscommunication at the receiver's end on the basis of sensors detecting noise in the communication. The system can then recommend to increase the level of explanations. More sensors transmit more information to users about the environmental conditions as well as the bodily conditions; most probably people will not ignore the information but rather feel pressurized to adapt. The new information technologies and infrastructures necessitate but also enable more adaptation.

The five-step procedure for designing systems that support adaptive behaviour relies on knowledge of the user, task and setting. The first step is to use the knowledge of the user's behaviour when solving the task in order to determine the dimensions of changing behaviour. More knowledge and more sophisticated computing and data analytics will allow building better support for adaptive behaviour.

References

Billsus, D., Brunk, C. A., Evans, C., Gladish, B., & Pazzani, M. (2002). Adaptive interfaces for ubiquitous web access. *Communications of the ACM*,*45*(5), 34-38.

Djamasbi, S., Siegel, M., & Tullis, T. (2011). Visual hierarchy and viewing behavior: An eye tracking study. In *Human-Computer Interaction. Design and Development Approaches* (pp. 331-340). Springer Berlin Heidelberg.

Kennedy, M.; Te'eni, D.; and Treleavan, J. (1998). Impacts of decision task, data and display on strategies forextracting information. *International Journal of Human Computer Studies,* 48, 159–180.

Louis, M. R., & Sutton, R. I. (1991). Switching cognitive gears: From habits of mind to active thinking. *Human relations*, *44*(1), 55-76.

Merminod, V., Rowe F., Te'eni D (2012) Knowledge Sharing and Maturation in Circles of Trust: The case of New Product Development ICIS2012, Orlando, FL, December.

Ronen H. and Te'eni D (2013). The Impact of HCI Design on Health Behavior: The Case for Visual, Interactive, Personalized (VIP) Feedback. ICIS2013, Milan, December, 16-18, 2013

Srinivasan A & Te'eni D (1995). Modeling as constrained problem solving: An empirical study of the data modeling process. *Management Science*, 41(3), 419-434.

Te'eni, D. and Sani-Kuperberg, Z (2005). Levels of abstraction in designs of human-computer interaction: The case of e-mail. *Computers in Human Behavior*, 21, 817-830.

Vessey, I., & Galletta, D. (1991). Cognitive fit: An empirical study of information acquisition. *Information systems research*, *2*(1), 63-84.